

AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior claim listings and versions.

1-75. (Canceled)

76. (Currently Amended) A method of automatically modifying an executable file comprising the steps of:

~~pseudo-randomly permuting a sequence of instructions within the executable file;~~
identifying one or more instructions ~~of the sequence of instructions and/or one or more variables, said instructions and/or variables at pseudo-randomly chosen locations~~ within the executable file;

inserting data and/or one or more instructions within said executable file before or after said identified one or more instructions and/or one or more variables, whereby any inserted instruction implements license verification code and any inserted data is license related; and

relocating all instructions and all variables within said executable file affected by the insertion, and adjusting all references to the relocated instructions and/or variables to reflect the relocating of the affected instructions and variables, wherein the identification of the one or more instructions and/or one or more variables is based on information obtained from an initial or an intermediate state of the creation process resulting in the executable file.

77. (Previously Presented) The method according to claim 76, characterized in that at least part of the data and/or the one or more instructions inserted within the executable file enable automatic testing of the integrity of at least one part of the executable file.

78. (Canceled)

79. (Currently Amended) The method according to claim ~~[[78]]~~76, characterized in that the information is obtained from object files created in the process of generating the executable file from a source code file.

80. (Canceled)

81. (Currently Amended) The method according to claim ~~[[78]]~~76, characterized in that the information is obtained from relocation information created in the process of generating the executable file from a source code file.

82. (Previously Presented) The method according to claim 76, characterized in that the data and/or the one or more instructions inserted in the executable file enable an identification of a licensee of the software product comprising the executable file.

83. (Previously Presented) The method according to claim 76, characterized in that the data and/or the one or more instructions inserted in the executable file enable an identification of the executable file itself.

84. (Previously Presented) The method according to claim 76, characterized in that the data and/or the one or more instructions inserted in the executable file enable an identification of a master file from which the executable file forms a copy before being modified.

85. (Previously Presented) The method according to claim 76, characterized in that the one or more instructions inserted in the executable file create a query to an execution control software program for a permission to run the executable file, and the one or more inserted instructions control the execution of the executable file in accordance to the permission being granted or denied.

86. (Previously Presented) The method according to claim 76, characterized in that the one or more instructions inserted in the executable file monitor changes to the

executable file and the one or more inserted instructions create a message indicating an infringement of the integrity of the executable file upon a change not being verified.

87. (Previously Presented) The method according to claim 85, characterized in that granting the permission to run the executable file comprises validation information in form of a request ticket.

88. (Previously Presented) The method according to claim 85, characterized in that the permission to run the executable file is formed by a runtime ticket.

89. (Previously Presented) The method according to claim 85, characterized in that the one or more instructions inserted in the executable file receive a log-off ticket and the one or more inserted instructions insert the log-off ticket within the executable file.

90. (Previously Presented) The method according to claim 89, characterized in that the one or more instructions inserted in the executable file return the log-off ticket to the execution control software program upon terminating the execution of the executable file.

91. (Previously Presented) The method according to claim 87, characterized in that the one or more instructions inserted in the executable file comprises verification code for verifying the validity of at least one type of ticket.

92. (Previously Presented) The method according to claim 85, characterized in that the one or more instructions inserted in one or more instructions of the executable file increment a counter related to the respective instruction each time said instruction of the executable file is involved.

93. (Previously Presented) The method according to claim 92, characterized in that the one or more instructions inserted in one or more instructions of the executable file send data concerning the value of the counter to the execution control software program upon terminating the execution of the executable file.

94. (Previously Presented) The method according to claim 87, characterized in that the one or more instructions inserted in the executable file comprises means for an execution of code received from the execution control software program.

95. (Previously Presented) The method according to claim 94, characterized in that the one or more instructions inserted in the executable file comprises means for returning a result of the execution of said code to said execution control software program.

96. (Previously Presented) The method according to claim 76, characterized by at least one of changing an arrangement of at least two subroutines and changing the arrangement of at least two variables within the executable file.

97. (Previously Presented) The method according to claim 96, characterized in that the changing of the arrangement of the at least two subroutines and the changing of the at least two variables is performed by a pseudo-random permutation.

98. (Currently Amended) A processor-readable medium incorporating a program of instructions configured to perform a method of automatically modifying an executable file, said method comprising:

~~pseudo-randomly permuting a sequence of instructions within the executable file;~~
~~identifying one or more instructions of the sequence of instructions and/or one or more variables, said instructions and/or variables at pseudo-randomly chosen locations~~
within the executable file;

inserting data and/or one or more instructions within said executable file before or after said identified one or more instructions and/or one or more variables, whereby any inserted instruction implements license verification code and any inserted data is license related; and

relocating all instructions and all variables within said executable file affected by the insertion, and adapting all references to the relocated instructions and/or variables to reflect the relocating of the affected instructions and variables, wherein the identification of the one or more instructions and/or one or more variables is based on information obtained from an initial or an intermediate state of the creation process resulting in the executable file.